ARI Research Note 87-51

HUMAN PROBLEM SOLVING IN DYNAMIC ENVIRONMENTS: UNDERSTAINDING AND SUPPORTING OPERATORS IN LARGE-SCALE, COMPLEX SYSTEMS

Richard L. Henneman
William B. Rouse
Georgia Institute of Technology

for

Contracting Officer's Representative
Michael Drillings

BASIC RESEARCH LABORATORY
Michael Kaplan, Director

U. S. Army
Research Institute for the Behavioral and Social Sciences

October 1987

Approved for public release; distribution unlimited.

U. S. ARMY RESEARCH INSTITUTE

FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the

Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON **Technical Director**	**WM. DARRYL HENDERSON** **COL, IN** **Commanding**

Research accomplished under contract
for the Department of the Army

Georgia Institute of Technology

Technical review by

Michael Kaplan

This report, as submitted by the contractor, has been cleared for release to Defense Technical Information Center (DTIC) to comply with regulatory requirements. It has been given no primary distribution other than to DTIC and will be available only through DTIC or other reference services such as the National Technical Information Service (NTIS). The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARI Research Note 87-51	2. GOVT ACCESSION NO. AD-A189	3. RECIPIENT'S CATALOG NUMBER 539
4. TITLE (and Subtitle) Human Problem Solving in Dynamic Environments: Understanding and Supporting Operators in Large-Scale, Complex Systems		5. TYPE OF REPORT & PERIOD COVERED Final Report June 82 - December 86
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard L. Henneman and William B. Rouse		8. CONTRACT OR GRANT NUMBER(s) MDA903-82-C-0145
9. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Man-Machine Systems Research Georgia Institute of Technology Atlanta, GA 30332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2Q161102B74F
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Research Institute for the Behavioral and Social Sciences, 5001 Eisenhower Avenue, Alexandria, VA 22333-5600		12. REPORT DATE October 1987
		13. NUMBER OF PAGES 72
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) - -		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE n/a

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

- -

18. SUPPLEMENTARY NOTES

Michael Drillings, contracting officer's representative

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Man-Machine,
Computer Models,
Human Performance,
Communication Networks.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This research note seeks an empirical understanding of the relationship between the physical characteristics of a large-scale system and human performance, and formalizing these empirical results into several measures of large-scale complexity. Behavioral computer models of the human operator in a large-scale environment were then constructed, and the models were found consistent with human behavior. This approach showed subtle performance improvement for aided subjects. Keywords:

HUMAN PROBLEM SOLVING IN DYNAMIC ENVIRONMENTS: UNDERSTANDING AND SUPPORTING OPERATORS IN LARGE-SCALE, COMPLEX SYSTEMS

Richard L. Henneman
William B. Rouse

ABSTRACT

Human performance in monitoring and controlling large-scale, dynamic systems is considered. Initial efforts were directed at obtaining an empirical understanding of the relationship between the physical characteristics of a large-scale system and human performance. Results showed the very strong effect that number of levels and degree of interconnectedness can have on human performance. Later efforts formalized these empirical results into several measures of large-scale system complexity. Two dimensions of complexity were proposed, measured, and evaluated: structural complexity results from the physical structure of the system, while strategic complexity results from an operator's understanding of the system. The knowledge gained from these engineering approaches was used to develop a behavioral model of the human operator in a large-scale environment. A comparison of the model's performance to human performance indicated that the model was consistent with human behavior. The model was then used to provide cognitively plausible decision aid to the human operator. Results from this approach were promising in that they showed subtle performance improvement for the aided subjects.

I. INTRODUCTION

Recent trends in automation have facilitated the creation of large, complex engineering systems, such as communication networks [Henneman and Rouse 1984b, 1986]. These systems are frequently represented as hierarchical networks that consist of a very large number of nodes connected by arcs. The network size and degree of integration combine to create systems of potentially enormous complexity. Because of this complexity, computer technology is often used to control the systems. For example, the functions of a large-scale command and control network may depend on a distributed set of intelligent control systems.

Although full automation often is appropriate during normal situations, control is likely to be transferred to a human operator during abnormal or infrequent events, e.g., system failures. In this way, the manner in which people interact with large *systems is fundamental*ly being changed. The human operator is no longer in charge of the routine, continuous control of the system. Rather, the operator is mostly concerned with the unexpected, the unusual, and the non-routine aspects of system control. Requisite human skills for system control are shifting from psychomotor to problem solving [Wickens 1984].

The use of automation in control systems during normal operation raises questions about the human's ability to control the system during abnormal situations. Since operators infrequently interact with the system, their knowledge of the system dynamics, structure, and context may be inadequate to cope with the complex task demands of abnormal situations. This problem is especially critical in dynamic environments in which the state of the system is time-varying; timely resolution of crisis situations is dependent upon the human's ability to retrieve and use relevant task knowledge quickly. In addition, in a context-rich environment the human's problem-solving performance is also dependent on the human's internal representation of that contextual knowledge.

This report considers human performance in monitoring and controlling large-scale systems by reporting the methods, results, and conclusions of a series of four experiments within a particular environment. Initial efforts were directed at obtaining an empirical understanding of the relationship between the characteristics of a large-scale system and human performance. Later efforts formalized these empirical results into several measures of large-scale system complexity. Finally, the knowledge gained from these engineering approaches was used to develop a behavioral model of the human operator in a large-scale dynamic environment. This model formed the basis of an approach to aiding the human operator. The report roughly follows this chronology.

II. SYSTEM CHARACTERISTICS AND HUMAN PERFORMANCE

A. Characteristics Of Large-Scale Systems

When considering human performance in interacting with a complex system, difficulty often arises in trying to exercise adequate experimental control over the characteristics of the real system. A variety of exogenous variables may mask the true effect of the variable of interest. Moreover, due to cost constraints, it is often difficult to make the changes in system characteristics necessary to elicit variations in human performance. For example, if in a large-scale system the variable of interest is the number of levels in the system, it is infeasible to alter the structure of a real system. Thus, a common approach to studying human performance in interacting with a complex system is to use a computer-based simulated abstraction of the real system.

Two simulations, MABEL and CAIN, were developed for the purposes of this study. MABEL is a relatively context-free representation of a large-scale system in which a human operator is required to monitor and control the real-time functioning of the system. By issuing commands, the operator accesses and displays activities within various parts of the hierarchical system, acquires information about the current system state, and issues control actions (e.g., component repairs and load shedding) when required. An operator's major task in the simulations is to diagnose and repair failed components. The CAIN system is structurally isomorphic to MABEL; the difference is that CAIN is contextually augmented, representing the nationwide telephone system.

In order to provide a context for the simulations, experiments, and results presented in this report, the next section describes the general features of two existing large-scale systems, the nationwide telephone system of AT&T and the U.S. Army´s Communications-Electronics Management System. The purpose is to present physical characteristics of these real systems and to describe ways in which people interact with the systems.

1. Examples of Large-Scale Systems

The nationwide telephone system [AT&T; Ash and Mummert 1984; Mocenigo and Tow 1984] has functioned until recently as a five-level hierarchical network composed of more than 170 million telephones and more than 22000 switching centers. The network consists of two basic elements: transmission and switching. The transmission elements are the actual communication paths that messages take through the system; the switching stations serve to interconnect calls economically.

A major feature of the system is its high degree of automation. Messages are sent through the system hierarchy via direct or alternate paths that have been pre-determined. The system operates under normal conditions without any manual intervention. The switching stations, serving as repositories of network intelligence, automatically perform such tasks as 1) determination of source, destination, and path through the network; 2) testing of lines for busy conditions before establishing a path; and 3) continual checking of circuit conditions.

Nonetheless, human monitoring and maintaining of the system is still a necessity. During overload situations or in the case of major equipment failures, network performance can degrade rapidly. Human network controllers must intervene when the automatic solutions are excessively expensive or when a problem arises requiring human judgement. To deal with these situations, two general categories of control exist -- expansive and protective [Ash and

Mummert 1984]. Expansive controls increase the network capacity by providing substitute or alternate routes for calls that are blocked. Protective controls reduce the number of calls entering a congested portion of the network or reduce the number of routing alternatives. Thus, the human operator can implement these controls by cancelling alternate routing, rerouting calls, issuing line load controls, and playing recorded announcements.

Recently the national phone network of AT&T altered its structure considerably. Instead of being structured hierarchically as explained above, a new approach called Dynamic Nonhierarchical Routing (DNHR) is being used [Ash and Mummert 1984]. The system is termed dynamic because a call may be routed over different paths at different times of the day to take advantage of spare network capacity. The system is termed nonhierarchical because switches are no longer separated into a hierarchy of different classes; they are equivalent in function. In short, any call may be routed through any part of the network to reach its destination.

As Mocenigo and Tow [1984] point out, managing the DNHR network is analogous to finding "a moving needle in a moving haystack" due to the increased dynamic nature of the system. Recent research efforts at AT&T have been directed towards introducing a higher degree of intelligence into the system, thus automating the system to an even greater degree. As Mocenigo and Tow note, however, it will not be possible to eliminate the role that the human monitor must play in this system, largely due to the problems that system failures create.

A very different type of large-scale system, the Army's Communications-Electronics Management System (C-EMS) [U.S. Army 1977a, 1977b], is designed as a means to meet the communications requirements of the battlefield. Due to the dynamic nature of its environment, the system lacks the permanence and the level of automation of the nationwide phone system. In addition, since the military must be mobile during combat operations, a high degree of engineering and planning is required to produce an integrated, effective system. During a battle, for example, parts of the network may be damaged or communication units may change locations. The system must be able to respond quickly to these changing resource capacities and network configurations.

The C-EMS is composed of several different forms of communication device. Although the telephone network described above is composed largely of phone-related equipment, the C-EMS network may consist of radio, wire and cable systems, radio-wire integrated systems, messenger services, and visual and sound communications. These system components typically are arranged hierarchically in a manner similar to that of the phone network. Each device is referred to as a node. The specific system requirements are dependent upon the type of information to be transmitted, the form in which it will be received, and the security and speed required. Thus, not only is the C-EMS more mobile and less automated than the nationwide phone system, but it can also be described as less homogeneous.

2. MABEL

MABEL (Monitoring, Accessing, Browsing, and Evaluating Limits) [Henneman and Rouse 1984b; Henneman 1985a] is programmed in Pascal on a VAX 11/780 computer and operates in real time. It is structured as a large network that can range in size from hundreds to thousands of nodes. Customers travel through the system from a randomly selected source node to a random destination. Subjects monitor this system activity via a CRT display. When they detect a problem in the system (possibly due to a failure), subjects issue an appropriate command through a keyboard to correct and compensate for the abnormal situation. The overall system objectives are: 1) to maximize the number of customers served, and 2) to minimize the time it takes for customers to travel between their source and destination nodes.

The following sections discuss MABEL in more detail. Emphasis is placed on the structure of MABEL, the operator interface, and typical system operation.

System structure. Several elements are basic to the structure of MABEL. A node represents the smallest structural unit in the network. Customers are processed at nodes with service times that follow an exponential distribution. Each customer is passed from node to node, following a path that will minimize its expected time in the system. If a node in a customer's path is currently busy, the customer joins a waiting line at that node until the node becomes idle.

As mentioned above, MABEL can consist of hundreds or thousands of nodes. It is impossible for the human operator to perceive and process information about all of the nodes at one time. On a more practical level, it is impossible to display information about all of these nodes at one time. Thus, nodes are grouped into relatively small networks called clusters. Human operators are restricted to viewing only one cluster at a time on the MABEL display.

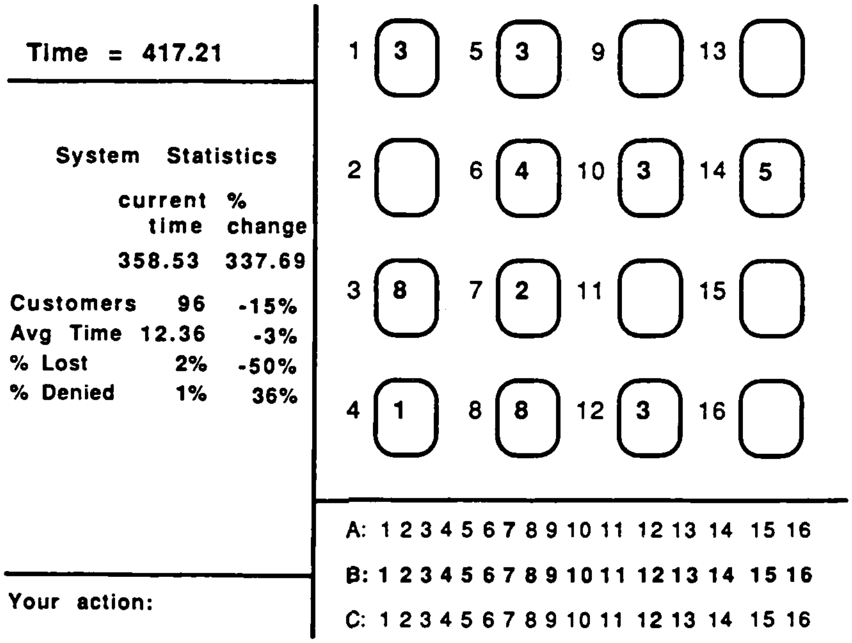
These clusters are grouped into hierarchical levels. A customer typically enters the system through a cluster in the lowest level. The customer proceeds through that cluster to a node that connects to a cluster in the next higher level. This process repeats until the customer reaches the top level of the system. At this point, the process reverses: the customer travels through a cluster and then "jumps" down to the next lower level. The process repeats until the customer reaches his destination.

Thus, as noted above, the system is analogous to a telephone communications system. Imagine a hypothetical three-level network in which a call is placed from Americus, GA to Mason City, IA. The message first travels from Americus to Macon to Atlanta via a network of telephone lines and switching stations. The message then travels from Atlanta to Chicago. It is then transferred to Davenport, IA, and finally proceeds to Mason City. Atlanta and Chicago are at the highest level of the hierarchical system; Macon and Davenport are at the second level, while Americus and Mason City are at the lowest level.

Customers initially arrive at a node in the lowest level in the system. These arrivals are scheduled according to a Poisson process. Routing through the system is completed automatically as determined by a shortest path algorithm. Thus, customers are routed through nodes that will minimize the time they spend in the system.

<u>Operator interface</u>. Subjects obtain information about MABEL from a video display (Figure 1). The screen is divided into several sections. The upper right portion of the screen displays a cluster of nodes. The dim numbers to the left of each node identify the node, while the numbers inside each node represent the current queue size (the number of customers waiting to be served). This portion of the screen is updated approximately every three seconds. A different cluster of nodes is viewed by entering an appropriate command.

The lower right portion of the screen is an aid to the user to identify the current displayed cluster. Each letter (A,B,C) represents a level in the hierarchy. Each number (1,2,3,...) represents either a node or a cluster.



```
Time = 417.21              1 [3]   5 [3]   9 [ ]  13 [ ]

    System  Statistics     2 [ ]   6 [4]  10 [3]  14 [5]
          current  %
            time   change
          358.53   337.69  3 [8]   7 [2]  11 [ ]  15 [ ]
Customers     96   -15%
Avg Time   12.36    -3%
% Lost        2%   -50%
% Denied      1%    36%    4 [1]   8 [8]  12 [3]  16 [ ]

                           A: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

                           B: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Your action:
                           C: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

Figure 1. MABEL Display

Bright and dim characters are used to indicate the subject's current position in the hierarchy. A row of characters that is completely bright represents the cluster that is currently displayed on the screen. One bright character in a row of characters indicates the node above the currently displayed cluster. In Figure 1, therefore, the displayed cluster is in Level B. This cluster is beneath Node 7 of Level A.

The upper left portion of the screen is used to display the current time. Time is updated approximately every three seconds. Since the system operates in real time, customers will keep arriving to the system whether any action is taken by the operator or not.

The middle left portion is used to display a variety of user-requested information about the system. This information is input at the prompt "Your action:", located at the lower left part of the screen. Ten different commands are available to the user. These can be grouped into four categories: access, monitor, diagnosis, and control. The ten different commands are summarized in Table 1.

Typical system operation. Under normal circumstances, MABEL will operate automatically without any interference from the human monitor. When a node failure occurs, however, the human must act to diagnose and repair it. Node failures in MABEL can occur in two ways:

ACCESS Commands	
d	down one level
u	up one level
MONITOR Commands	
m	monitor next level
s	system summary statistics
c	cluster summary statistics
DIAGNOSTIC Commands	
t	test displayed cluster for failures
n	node information
CONTROL Commands	
r	replace node
l	reduce load

Table 1. MABEL Command Set

(1) Total failure due to malfunctioning equipment: in this case a node is unable to service any customers waiting at it. All customers waiting at the node are lost, thereby reducing the queue size to zero. Additionally, the node refuses to accept any customers from another node. These customers are retained at their previous node. Since they are unable to proceed, the situation may lead to the following type of failure.

(2) Failure due to exceeding the capacity of the node: each node has a maximum number of customers that it can store at any one time -- that is, each node has a maximum queue size. If this queue size is exceeded, the node fails. Its behavior after this point is identical to equipment failures. The node is unable to accept customers and, thus, new customers are retained at their old node. Once a failure occurs, therefore, it is likely to lead to other failures. In the extreme case, if nothing is done to repair failed nodes, the entire system will fail.

It is, of course, also possible for this type of failure to be induced simply by trying to service too many customers, i.e., the system is trying to handle too much of the load. In this case, customers arrive at a node at a rate faster than the node can service them.

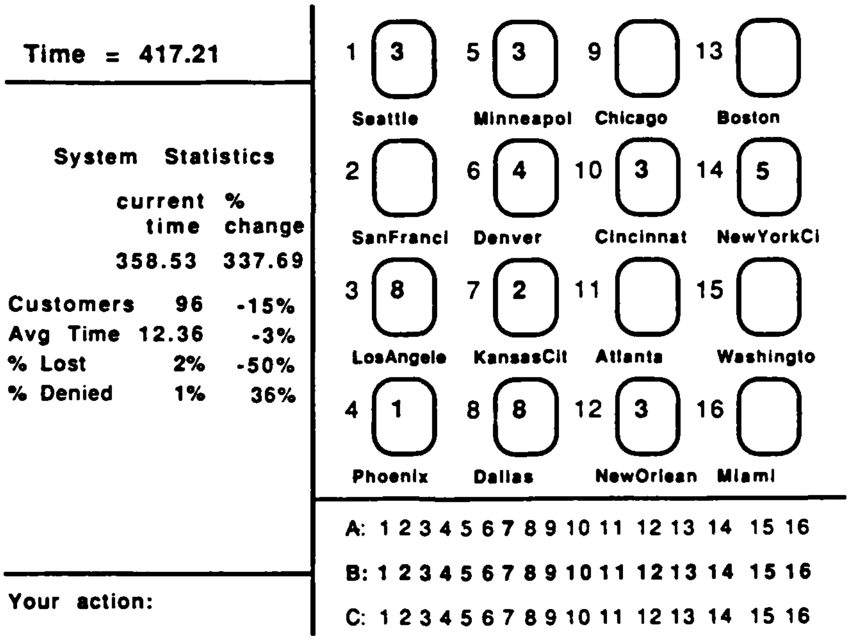
Subjects locate failures by monitoring critical system states and testing suspect nodes or clusters of nodes. If a failure is found, the subject dispatches a crew to repair the node. If the system becomes too crowded with customers, the subject can issue a command to reduce the number of customers admitted to the system.

3. CAIN

Certain features of MABEL were substantially changed to develop CAIN (Contextually Augmented Integrated Network) [Henneman and Rouse 1986; Henneman 1985a]; however, the underlying structure of CAIN is identical to that of MABEL. CAIN, however, is contextually augmented. The simulation has a much higher level of fidelity in that the addition of context produces a simulation with a much stronger resemblance to a real system.

Thus, although the physical hierarchical structure of MABEL was preserved, the addition of contextual information to CAIN required changing some features of the interface. In the MABEL scenario, for example, all nodes on a display page are identified by a number on the CRT display. Each displayed node in a cluster, therefore, is physically identical to nodes in other clusters. The MABEL interface has a generic quality in that all subsystems are visually similar; no contextual cues exist. On the other hand, nodes in CAIN are identified via specific geographic locations. Thus a node in MABEL with the label "9" might be labeled "Chicago" in CAIN. A typical CAIN display is shown in Figure 2.

Simply introducing geographic names as node labels is not enough, however, to alter subject task performance. A small experiment (n=3) replicated the first MABEL experiment, with the exception that nodes were given geographic names. Subjects still referred to nodes by number only; contextual labels were present but not needed to perform the task. No significant difference was found in terms of performance between subjects using the two task scenarios. This resulc suggests that the addition of context must be such that it provides associative links (i.e., memory aids) or cues (i.e., clues to the location of problems within the system) through which subject



```
Time = 417.21

System  Statistics
          current    %
             time  change
           358.53  337.69
Customers      96    -15%
Avg Time    12.36     -3%
% Lost         2%    -50%
% Denied       1%     36%

Your action:
```

```
A: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
B: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
C: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

Figure 2. CAIN Display

performance is enhanced or task difficulty is decreased.

Associative links. The formation of associative links in CAIN is facilitated by the way in which a subject identifies a node. In CAIN, nodes are referred to by geographic labels only, never by number. Subjects may input the shortest string of characters that uniquely identifies the node from all other nodes in the system. Thus, "Denver" may be abbreviated "den". Most nodes can be identified with a three or four character substring of the complete name. In addition, the number of elements on a display page is kept constant at 16 so that the contextual information is invariant.

To illustrate the effect this change has on the subject´s task, consider the command that displays a lower-level cluster. In MABEL, the subject inputs the command "d2," which displays the cluster beneath node 2. In CAIN, on the other hand, the subject types "dsanf," which displays the cities beneath San Francisco (e.g., Berkeley, San Jose). Thus, subjects can form associations or links between system parts due to the existence of contextual information.

Subjects can use these learned associative links to maneuver through the CAIN display hierarchy. In MABEL, movement between display pages is constrained to the cluster of nodes immediately above or below the current display. Thus, it is not possible to jump laterally across the network. In CAIN, however, it is possible to move from one part of the system to any other part. For example, if a subject recalls that the cluster associated with Bangor, Maine, was previously experiencing problems, it is simple to call up that particular cluster display. In addition, subjects can always return immediately to the highest level in the system.

Cues. The formation of cues in CAIN is provided by the introduction of context-dependent events. These events are of one of two types: recurring failures and nonuniform loading. Although equipment in nodes fails randomly, some equipment experiences a higher probability of failure. For example, a thunderstorm in Little Rock, Arkansas, may make equipment in that city susceptible to lightning damage. Similarly, given that incidents of vandalism are more likely to occur in Newark, New Jersey than Council Bluffs, Iowa, there is a greater chance of equipment damage in Newark. Therefore, equipment in certain cities exhibits a greater tendency to fail than in other cities. Subjects are informed of these locations via warning alarms that appear on the bottom of the display. Subjects can directly monitor activities within these trouble spots via a special "watch" command.

Besides recurring failures, another type of context-dependent event present in CAIN is nonuniform loading. At different times, some sections of the system may be prone to heavy loading. For example, certain times of day are busier in one part of the country than in others. Similarly, a major political or sports event in one section of the country may increase the number of messages sent. As with the recurring failures, subjects are told the location of these increased loads via a message at the bottom of the screen. Subjects can then reduce the number of customers admitted to the overloaded subsystem.

In summary, despite the structural ismorphism of the two simulations, CAIN represents a significant departure from the context-free scenario of MABEL. Through the addition of contextual detail and the addition of events that are dependent upon this contextual information, the simulation fidelity has been increased significantly.

4. Issues

Initial efforts in this research program were to gain an understanding of the relationship between physical characteristics of a system and human performance in monitoring and controlling such a system. For example, information displays for computer-based large-scale systems are frequently constrained by their size: only a limited amount of information may be displayed at one time. Thus, the number of elements of a system presented at one time may affect the ability of the operator to perceive relevant system state information both rapidly and accurately.

Another system characteristic that may affect human monitoring and control performance is the number of hierarchic levels. A system with multiple levels may have a very strong effect, for example, on the length of time needed for a human operator to find a failed component. Although some guidance exists within the literature relative to trade-offs between depth and breadth in static display menu hierarchies [Paap and Roske-Hofstrand 1986], little guidance exists for dynamic systems. Finally, another system characteristic of interest is the rate at which system components fail. If the main role of the human operator in a large dynamic system is to diagnose failures, an important issue is whether or not humans can change their control strategies to adapt to changes in the quality (or reliability) of individual system components. These system characteristics (display size, number of levels, and component failure rate) were considered in Experiment One.

B. Experiment One: Empirical Analysis

1. Method

Twelve volunteer subjects were exposed to MABEL by a set of written instructions [Henneman and Rouse 1984b]. These instructions contained a detailed explanation of the overall structure and normal operation of MABEL, a summary of the commands, and an explanation of the subject's role in operating MABEL during off-normal situations. Summary sheets of this information were also available. A quiz verified subjects' understanding of the effects of failures on system performance.

Training concluded with a special version of MABEL that allowed subjects to stop the execution of the program at any time during the experimental run. This training version of MABEL had the advantages of allowing subjects to become familiar with the commands and to become aware of the effects of failures on both display features and system performance without being overwhelmed by the progressive effects of failures. If a situation became too complex, the subject could simply halt the dynamic system, solve the problem, and proceed. Subjects supervised two different training scenarios: a system with 16 nodes/cluster and 2 levels, and a system with 9 nodes/cluster and 3 levels.

The experiment had three independent variables: cluster size (i.e., number of nodes per display) and number of levels functioned as within-subject factors and failure rate served as a between-subjects factor. Cluster size varied between 4, 9, and 16 nodes; number of levels varied between 2 and 3. Failure rate was defined as the probability that a randomly selected node in the system would fail during each iteration of the MABEL program. One iteration occurred after each activity in the network (for example, the arrival of a new customer to the system). Failure rate was either low (probability of failure/iteration = .0005) or high (probability of failure/iteration = .0010). The six subjects in each group controlled six systems corresponding to all possible combinations of cluster size and number of levels. The order of presentation to subjects was balanced in order to average out any residual training effect.

Performance was assessed in several ways. The measures can be broadly grouped into two categories, namely, product and process [Henneman and Rouse 1984a]. Product measures assess the final result of a problem-solving session (such as number of customers served) and, thus, assess system-human performance. Process measures, on the other hand, assess how that result was obtained by evaluating individual steps in a subject's strategic approach to supervising the system.

The product measures calculated the length of time customers spend in the system (mean sojurn time) and the number of customers served during an experimental run. (These measures were normalized to account for inherent differences that exist among the different experimental system configurations. Henneman and Rouse [1984b] provide a description of this bias-correcting procedure.) Process measures were classified into three types: 1) errors (e.g., number of times a subject viewed a failed node but did not repair it), 2) failure diagnosis (mean time to diagnose a failure and the fraction of failures found), and 3) strategy (e.g., mean amount of time spent accessing, monitoring, diagnosing or controlling).

2. Results[1]

Analyses of variance were performed to determine the effect of the independent variables (cluster size, number of levels, and failure rate) on each of the dependent measures. Overall trends within the product measures of performance were very consistent: performance degraded significantly with increasing number of levels and significantly improved with increasing display size. The effect of number of levels was very strong, producing up to a 5-fold degradation in level of performance. This effect was expected: the greater the percentage of nodes hidden from view, the greater the difficulty subjects experienced in supervising the system. For instance, the three-level systems resulted in substantially longer times to diagnose failures. (The mean time to failure diagnosis for the three-level systems was 76 s; the mean time to failure diagnosis for the two-level systems was 18 s.) Since it took

[1] Results are reported in greater detail in Henneman and Rouse [1984b]. All results mentioned here are significant at $p < 0.01$.

more time for the effects of lower-level failures to become obvious at the higher levels, the effects tended to be more serious than in the two-level systems. This lengthened diagnosis time tended to degrade most other performance dimensions.

A trend not predicted was that increasing cluster size would lead to improved performance. One would suspect that larger numbers of nodes per display should lead to increased task complexity. Thus, as the number of components that the human must deal with increases, performance should degrade. This is not the case with MABEL. A main reason for this result is that the larger systems are inherently more reliable than the smaller systems: the small systems contain fewer alternate paths between nodes through which customers can be rerouted following a failure. Thus, customers tend to be retained more frequently at nodes when they have fewer alternate paths through the system. This system characteristic also accounts for the shorter failure diagnosis times found in the small cluster size systems: failures and their symptoms propagate faster in the small systems. (The mean times to failure diagnosis for the 4, 9, and 16 node clusters were 24 s, 39 s, and 77 s, respectively.)

Failure rate did not play a role in shaping performance except with respect to the measures of strategy, e.g., the percent of time subjects spent performing different activities (accessing, monitoring, diagnosing, and controlling). Results suggested the prevalence of two strategies for supervisory control of MABEL. One strategy involved staying at higher levels and using monitor commands to assess the state of lower levels. The other strategy involved actually accessing the lower levels and performing tests to diagnose failures. Subjects with low failure rate conditions tended to select the former strategy, while subjects with high failure rate conditions tended to select the latter strategy. Apparently both strategies were effective in that performance was independent of failure rate. Thus, it appears that subjects could adopt strategies to compensate for decreased reliability of individual system components, but not for the more resource-constrained networks.

C. Experiment Two: Measuring Complexity

The initial experiment considered the relationship between several physical characteristics of a large-scale system and human performance. A second experiment addressed this relationship more quantitatively by evaluating several measures of task complexity. Based on a review of the literature [Henneman and Rouse 1986], two measures of complexity relevant to the task of human monitoring and control of large-scale systems were proposed. Two dominant perspectives were identified within the complexity literature, namely, that of the systems scientist and that of the behavioral scientist.

Most studies of complexity performed by systems scientists are on a context-free or theoretical level. Although much work has gone into defining and measuring system complexity, little has been done to assess the implications of complexity. Furthermore, while humans must play an important role in many large-scale systems (e.g., failure diagnosis and network management), little research has investigated the relationship between large-scale system complexity and human performance. Finally, due to the strong theoretical flavor of the systems science approach, it is often difficult to see its application to real-world systems.

On the other hand, studies of complexity performed by behavioral scientists are on a very applied level. Although the approach often lacks the mathematical rigor of the systems approach, complexity is always related to some aspect of human performance. Unfortunately, differences between tasks and complexity measures cause difficulty in generalizing results across contexts. Moreover, the small, well-defined nature of the tasks seems to have little relation to human performance in large-scale system.

The research described in the remainder of this section attempts to integrate several perspectives concerning the nature of complexity, as well as to illustrate the impact of this conceptualization of complexity on human performance in CAIN. Complexity is viewed as being a result of both the structure of the system and the human operator's understanding of the system. Complexity is also considered in terms of its relation to system performance and human performance. Thus, nonbehavioral and behavioral approaches are taken into account.

In this report, the complexity of a large-scale system is described in terms of: 1) the physical structure of the system and 2) operators' understanding of the system as reflected by their strategy. From this perspective, a system that is complex or difficult to control for one operator may be relatively easy to control for another operator. Similarly, the complexity of a system may vary with time for any particular operator. Some systems, however, may be complex regardless of any particular control strategy due to their inherent structural complexity. The following paragraphs propose two measures of complexity that incorporate these ideas. Structural complexity is considered first, followed by strategic complexity.

1. Complexity measures

Structural complexity. A one-to-one relationship exists between the simulated physical structure of CAIN and the actual structure of the display-page hierarchy. Since the main control task in CAIN is to locate failures, a measure of structural complexity should assess the difficulty of finding failures given the physical arrangement of the system. A major constraint placed on an operator's ability to locate failures is the hierarchical display structure; thus, it seems reasonable that structural complexity can be estimated by calculating the total number of display pages the operator must view in order to repair all system failures. Assuming that the operator knows the location of all failures, this measure represents the minimum number of pages necessary to find all system failures. Therefore, the structural complexity measure represents optimal performance given the constraints of the structure or arrangement of the system components. Operator performance affects this measure only in that individual operators may have more or fewer failures depending upon their fault finding ability.

To illustrate how this measure is calculated, consider the system in Figure 3. This hypothetical system contains four nodes per display page and has three levels. Each group of four rectangles represents a cluster of nodes (i.e., one display page). For clarity, only those clusters of nodes that enter into the complexity calculation are shown. The darkened rectangles represent nodes that have failed. In this example, therefore, three failures exist within the system: two on the second level and one on the third level.

The structural complexity measure is determined by counting the number of display pages that must be viewed in order to find all failures. The counting method assumes a strategy based on tracing higher level symptoms to their lower level causes. (Context-specific cues might, of course, allow operators

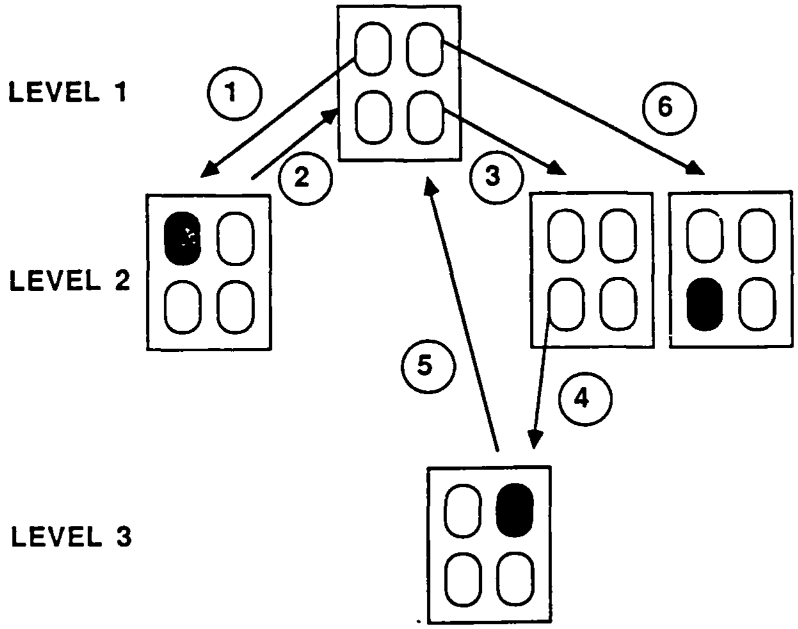


Figure 3. Example calculation of structural complexity.

to locate failures in fewer pages.) Thus, the counting method assumes that after locating all failures along one subsystem branch, the subject returns to the highest system level to search the next branch (a depth-first strategy). Figure 3 is self-explanatory: to repair all three failures in the system, an operator must view at least six display pages. The final return to the top system level is not counted into the measure because it would simply add one to all estimates.

Strategic complexity. The strategic complexity measure explicitly considers operator performance. When operators are deciding which path through the system is most likely to lead to finding a failure, they make a tradeoff between their uncertainty concerning the state of a subsystem display page (i.e., queue lengths) and their expectations of finding a failure in that subsystem. High uncertainty about a subsystem may be acceptable, for example, if a relatively low probability exists of finding a failure on that display page. On the other hand, high subsystem uncertainty may be unacceptable if a very high probability exists of finding a failure.[2]

State uncertainty U is defined as the real time elapsed since a particular display page was last tested for failures. Probability of failure is defined as the probability that a failure exists within a cluster, given the state of the display X, and is denoted by $p[F|X]$. For example, when a subject views a particular display page, features of that display provide information about the existence of failures in other subsystems (e.g., a large queue size suggests a lower level failure.) Experimental data files were replayed in order to estimate these probabilities empirically. These probabilities were determined by dividing the frequency with which a display state reflected a failure by the frequency with which a particular display state (i.e., queue length) was viewed by an operator. Sets of probabilities were calculated for different system configurations (2 vs. 3 levels and high vs. low redundancy) and different loading rates (e.g., a system with a low loading rate has fewer customers in service and, hence, lower threshold or queue size will reflect failures).

[2] Of course, the acceptance of uncertainty will also vary as a function of the consequences of a failure. If, for example, a failure is likely to lead soon to another failure, high uncertainty about that subsystem would be unacceptable. Subjects, however, do not have any knowledge of these possibly unequal probabilities. Thus, it is reasonable to assume equal effects of failures for this discussion.

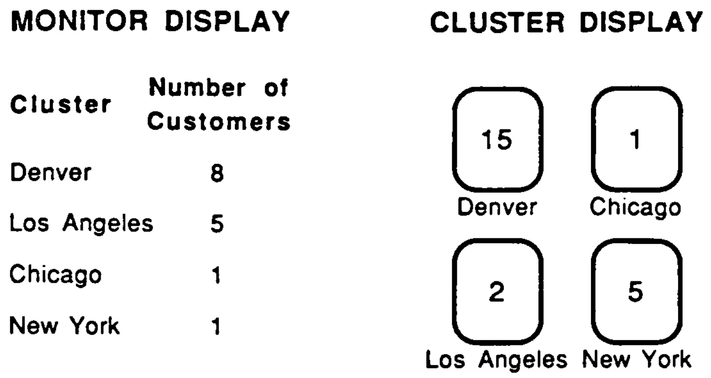
The measure of strategic complexity multiplies these two measures (state uncertainty and probability of failure given the system state) and sums the product across all clusters in the system:

$$\text{strategic complexity} = \sum U(i) * p[F|X(i)]$$

where U(i) is the time since last accessing display page i; X(i) is the state of page i reflected by the display one level higher; p[F|X(i)] is the probability of failure given state i; and F denotes "failure." Strictly speaking, this conceptualization results in strategic complexity having units of seconds. The U(i) values are really just "proxy" measures [Keeney and Raiffa 1976] of complexity, however, and thus, strategic complexity is left unitless.

When a subject descends to a lower level, the p[F|X(i)] remain fixed for the previous level. When a subject returns to the higher level, the p[F|X(i)] value associated with the just-visited lower-level clusters is set to zero. Thus, when an operator descends to a lower-level subsystem and tests for failures, the strategic complexity measure is simultaneously increased by the "new" uncertainty (i.e., increased U(i)) present in the other lower-level subsystems and decreased by the certainty (i.e., U(i) = 0) now associated with the current level.

To illustrate how the strategic complexity measure is determined, consider the display in Figure 4. This system contains four nodes per display page and has two levels. The operator is viewing the highest level page in the display hierarchy and is monitoring activity in the next level of the system. The operator can gather information about activity in the second level of the system from two sources in this example: the cluster display and the data displayed via the monitor command. The monitor command lists the number of customers in the clusters one level below; the cluster display shows the



MONITOR DISPLAY

Cluster	Number of Customers
Denver	8
Los Angeles	5
Chicago	1
New York	1

Figure 4. Example monitor and cluster display for calculation of strategic complexity.

number of customers waiting at all nodes in the current cluster.

Each of these pieces of information reflects the probability that a failure has occurred in a lower level cluster. These probabilities (which are plausible but hypothetical) are listed in Table 2. For example, the queue size of 15 in Denver reflects a relatively high probability (0.75) that a failure exists in level two. Similarly, the monitor command reports that eight customers are currently in the cluster beneath Denver; these eight customers reflect a 0.60 probability that a failure exists. The operator has not tested the cluster beneath Denver for failures for U(Denver) = 20.12 s. Using the information that reflects the highest probability of failure (i.e., from the cluster display) results in the following measure of strategic complexity for the Denver region:

$$U(\text{Denver}) * p[F|x(\text{Denver})] = 20.12 * 0.75$$
$$= 15.09$$

This procedure is then repeated for the other clusters in the network and the measures are added together. In this way the total strategic complexity is determined to be 15.61.

In this example, it should be noted that Denver makes a large contribution to the strategic complexity measure for two reasons: first, the operator has a high degree of uncertainty concerning the Denver subsystem in that it has not been tested for failures in 20.12 s. Second, the display reflects a very high probability (0.75) that a failure exists in the Denver subsystem. The combination of these two factors leads to a very high measure of strategic complexity for the Denver subsystem. On the other hand, the other subsystems have either a low uncertainty measure or a low probability of failure. Thus, as shown in Table 2, their contribution to strategic complexity is small.

Dependent measure of complexity. The literature review [Henneman and Rouse 1986] also suggested that an appropriate dependent measure of complexity is the time to failure diagnosis. In the context of CAIN, this measure is the mean time from when a failure occurs to when the subject issues a repair command for a failed node. Since the two independent complexity measures vary with time and since there are multiple repairs occurring in conjunction with the assessment of the variables, it was necessary to use a dependent measure that also changes with time. Average time, therefore, includes the diagnosis time for the current repair plus diagnosis times for the four previous repairs.

To summarize, the structural measure reflects an inherent characteristic of the network, namely, the number of display pages necessary to find all the failures in the system. The strategic measure, on the other hand, reflects temporal aspects of subjects' strategies, i.e., subjects' paths through the network. From this perspective, the strategic measure reflects the complexity resulting from a particular strategy.

Cluster	U	$p[F\mid X]$		$U \times p[F\mid X]$
	uncertainty	monitor	cluster	
Denver	20.12	.600	.750	15.090
Los Angeles	0.54	.100	.015	.054
Chicago	7.36	.001	.001	.007
New York	9.12	.001	.050	.456
		Strategic Complexity	=	15.607

Table 2. Example Calculation of Strategic Complexity

Although the two complexity measures proposed here may have some general applicability (in particular, the measure of strategic complexity is appealing due to its temporal nature), it is not the intent of this work to suggest or prove that these measure are true indices of task complexity. The goal instead is to show in a pragmatic sense that these two dimensions represent a useful distinction relative to task complexity. These measures are a convenient means to demonstrate this distinction.

2. Method

The main goal of Experiment Two was to investigate the nature of complexity in a large-scale human-machine system. As emphasized in the preceding section, the general assumption is made that task complexity can only be measured relative to an individual's understanding of the system and expertise in dealing with problems in that system. Thus, complexity is considered to be dynamic, varying across time and among subjects. Accordingly, as discussed below, subjects were required to perform the task (CAIN) over a relatively long period of time.

Results from the Experiment One indicated that cluster size (number of nodes per display page) in MABEL had a particularly strong effect on subject task performance. Results suggested that small clusters degraded performance because fewer connections existed between nodes; less redundancy caused failures to propagate more quickly. Another result from Experiment One showed the very strong effect of number of hierarchical system levels on human performance. Thus, two independent variables selected for further analysis were the degree of redundancy (or connectivity among components) and the number of levels in the system. (Cluster size was kept constant at 16 in order to emphasize the nonvarying features of the contextual display.) Redundancy or connectivity was defined as the number of connections emanating from each node. Redundancy varied between low (six connections between nodes) and high (13 connections per node), and the number of levels varied between two and three.

Of interest in this experiment was the way in which complexity changes as subjects gain expertise. Thus, the order of presentation of experimental conditions was not randomized. All subjects saw the same experimental conditions in the same order. A final independent variable, therefore, was the order of presentation of experimental conditions within each combination of number of levels and redundancy.

Eight paid subjects were trained in three sessions via a combination of written instructions and hands-on experience with CAIN, similar to that used in Experiment One. Subjects completed the first two training sessions by controlling a two-level CAIN system. The third training session was spent controlling a three-level CAIN system. As in Experiment One, these sessions were performed using a version of CAIN that allowed subjects to start and stop the program execution.

Summarizing the ten experimental sessions (S1-S10), they were performed in the following order (with the intent of increasing experimental difficulty): S1 and S2 had two levels with high redundancy; S3, S4, and S5 had three levels with high redundancy; S6 and S7 had two levels with low redundancy; and S8, S9, and S10 had three levels with low redundancy. Each experimental session was performed on consecutive days and lasted about 45 minutes.

3. Results[3]

Summary of Approach. Data from this experiment were first analyzed using the same performance measures used in Experiment One, e.g., mean time to failure diagnosis and fraction of failures repaired. Overall results from the analysis of variance of subject performance measures supported those of Experiment One. Measures of fault diagnosis performance were affected as expected by the independent variables. Increasing the number of system levels from two to three corresponded to a higher mean time to failure diagnosis (24 s vs. 50 s). This result was largely because failures take longer to propagate upwards in the three-level systems. In addition, failure-related symptoms take longer to emerge in highly interconnected networks; thus, the high redundancy systems resulted in longer mean times to diagnosis (41 s vs. 35 s). The fraction of failures repaired by subjects was also significantly affected by increasing the number of levels: as the number of levels increased from two to three, the fraction of failures found decreased from 0.95 to 0.69. As in Experiment One, subjects had difficulty coping with the very large search space in the three-level systems.

The data were also analyzed with the purpose of investigating relationships between the complexity measures, the CAIN environment, and operator performance. This investigation was accomplished in two ways. First, an analysis was undertaken of average or global measures of complexity (i.e., the complexity time series averaged over each experimental run). The effect of the experimental independent variables (number of levels and degree of inter-

[3] Results described in this section are reported in greater detail in Henneman and Rouse [1986]. All results are significant at least at the $p < 0.01$ level.

connectivity between nodes) on the mean complexity measures was determined using analysis of variance. The relationship between the mean complexity measures and measures of subject failure-diagnosis performance was then assessed by using correlation analysis. As is discussed below, this analysis of mean complexity values provided explanations for differences that exist between different system configurations.

The second way in which complexity was investigated involved using a fine-grained approach, namely, time-series analysis. Time-series analysis was selected due to the intrinsic time-varying nature of the independent and dependent complexity measures. This analysis provided insight into the way in which complexity evolves and affects different phases of the failure-diagnosis process.

Due to the amount of time. necessary to perform these analyses, the results are limited to Sessions 2, 5, 7, and 10. Data for the analyses were generated by replaying subject data files. Every three seconds (corresponding to the rate of display update), both complexity measures and the mean time to failure diagnosis were calculated. Mean values for all measures were calculated from these time series.

Analysis of average complexity measures. The results of two ANOVAs using mean structural and strategic complexity measures as dependent measures and number of levels and degree of redundancy as independent measures are summarized in this section. Structural complexity, as measured here, was significantly decreased in two ways: 1) decreasing the number of system levels and 2) decreasing the number of system failures. The first way enables subjects to access fewer display pages in order to diagnose failures in the lowest system level. The second way is facilitated by increasing the network redundancy (i.e., increasing the number of connections between nodes). As network redundancy increases, the mean number of node capacity failures decreases, which has the effect of decreasing the structural complexity measure.

Strategic complexity, as measured here, may be decreased in three ways: 1) using an effective strategy in terms of responding to symptoms, 2) decreasing redundancy, and 3) decreasing number of levels (which causes symptoms to emerge more rapidly). Subjects tended to trace failures to the lowest system level only when a symptom (i.e., visual cue) appeared on the display, even if they had not viewed a particular region in a long period of time. Consequently, when symptoms emerged slowly (as in the high-redundancy/three-level conditions), high uncertainty resulted. This uncertainty helped to create moderate to high strategic complexity. On the other hand, symptoms emerged more rapidly in the low-redundancy/two-level conditions. Since operators tended to wait for symptoms to emerge on the top-level display, low redundancy led to low values of strategic complexity.

This dependence on visual cues has implications for the design of task performance aids. One possibility is to have aids that help people to overcome their inability or reluctance to reduce system uncertainty despite the absence of failure symptoms. Alternatively, failure-related cues or symptoms could be enhanced so that operators naturally pursue leads sooner.

These results provide insight to the overall characteristics of the two complexity measures and their relationship to subject fault diagnosis performance. The measures are sensitive to variations among the system characteristics of number of levels and degree of redundancy. In general, the more complex systems have three rather than two levels. Although multiple system levels might be desirable in that they allow supervision of large networks and protect upper levels from the effects of failures, they have the undesirable side effect of masking symptoms from operators, thereby increasing the complexity of failure diagnosis. Multiple displays could possibly be used to reduce this complexity. The effect of redundancy on complexity depends on the type of complexity: more redundant systems (corresponding to less structural complexity) enhance the proper operation of the system by reducing the impact of failed components. On the other hand, more redundancy leads to increased strategic complexity (the complexity of failure diagnosis) due to the slower emergence of failure symptoms.

Beyond the characteristics of these single complexity dimensions is another important conceptual and methodological issue: the multidimensional nature of complexity, i.e., the relationship between the independent and dependent measures of complexity. A correlation analysis between the two average complexity measures and the two independent measures indicated that when many failures exist in a system, the general tendency is for the complexity measures to increase. At the same time, however, the mean time to failure diagnosis decreases. Thus, even though complexity may be large, failure-diagnosis time may be small.

This observation emphasizes the distinction mentioned previously between proper system functioning and the complexity of failure diagnosis. In a localized sense, control in a complex system is simple: no matter what the operator does, it will result in finding a problem (as reflected by short diagnosis times). In a global sense, however, control in a complex system is complex: so many problems may exist in the system that proper operation is endangered, as reflected by a low fraction of failures found. The operator, dealing with only a small part of the system at one time, may be oblivious to the scope of problems in the network. Another important issue is, therefore, the impact of a richly interconnected multiple-level system (that supports proper system functioning) on the complexity of human monitoring and control (that will degrade failure diagnosis performance).

Analysis of fine-grained complexity measures Time-series analysis [Fox and Jenkins 1976] was used to identify, estimate, and diagnostically check transfer functions that relate the two input complexity measures (structural and strategic) to the mean time to failure diagnosis. Each transfer function model predicted the current mean time to failure diagnosis through a linear combination of the complexity measures at various time lags. The essence of the modeling process was to determine the time lags to include in the model and the weight or relative contribution of each time-lagged variable to the predicted value.

Overall, the approach was successful. The equations removed all structure from the autocorrelation function of the model residuals. Furthermore, a comparison of the sum of squares of the original dependent time series (i.e., mean time to failure diagnosis) to the sum of squares of the residuals showed that the transfer functions explained 82 to 97 percent of the variance within the original data. Nevertheless, wide differences in the lag and coefficient values in the models existed among both subjects and systems.

A plausible explanation for these differences was derived by identifying certain characteristics of the task, the system, and the human related to the the process of failure diagnosis. For example, several different events are associated with the life cycle of each system failure: failure occurrence, symptom emergence, and failure diagnosis. Failure occurrence is when a part of the system fails. Symptom emergence is the time period between failure occurrence and the time a failure first affects any node that appears on the subject's video display. Failure diagnosis is the time period from failure occurrence to when a subject issues a repair command for a failed component. The timing of these events undoubtedly has some effect on the length of time needed to find the failure. Moreover, the system complexity at these event times might also affect failure diagnosis time.

Besides the possibility that different events associated with the failure life-cycle affect diagnosis time, it is also reasonable that different types of diagnosis might affect failure diagnosis time. The diagnosis of any particular failure may be classified as one of three types: topographic, symptomatic, or serendipitous. Subjects identifying failures using a topographic strategy trace failure symptoms from higher system levels to their causes in lower levels. Subjects identifying failures using a symptomatic strategy make a direct mapping from their knowledge of the system structure to the failed component. A symptomatic diagnosis relies, therefore, on the subject's contextual knowledge of the system. For example, when subjects make a jump from one cluster to another in the same level to repair a failure, their action suggests that their context-specific knowledge of the system is providing guidance to system trouble areas. Finally, subjects may also identify failures accidentally or serendipitously. In this diagnosis mode, subjects locate failures while browsing through the system or while tracing the cause of a different failure.

In summary, it is possible that several different types of failure-related event (e.g., failure occurrence and symptom emergence) and several different modes of failure diagnosis (e.g., symptomatic, topographic, and serendipitous) affected the time to failure diagnosis within a system. In addition, since the dependent complexity measure, mean time to failure diagnosis, was calculated based on an averaging window of five failure diagnosis times, it is possible for many lags (possibly quite long) to have entered the transfer function. From the perspective offered in the preceding paragraphs, therefore, the transfer functions relating the two complexity measure to failure diagnosis time were affected by types of failure-related event, modes of failure diagnosis, and the way in which diagnosis times were aggregated.

These factors were considered analytically by replaying subject data files and comparing measures of the characteristics described above to the transfer functions. Results showed that the variables and lags present in the transfer function were reasonable, if not entirely explainable. The real-time values of lags frequently agreed with the mean failure inter-event times calculated from subject data files. A comparison of these values suggested that recurring patterns of agreement existed between the lags and inter-event times. These recurring patterns were useful to explain the presence of both positive and negative terms in the transfer functions. Differences between time values can probably be accounted for by any of several reasons, including the high variability present within the data, the subjective nature of the modeling process, and the existence of events other than failure occurrence or symptom emergence (e.g., diagnosis time for a particular system level or subsystem) that affected parameters in the transfer functions.

These results demonstrate how two different dimensions of complexity, structural and strategic, can be related to human fault-diagnosis skill in a large-scale system. The exact nature of the two measures is relatively unimportant beyond a certain degree of intuitive validity. The importance of these results, however, lies in the demonstration that the complexity measures were dependent upon the number of failures in the system and the rate at which their symptoms emerge. These factors were highly dependent upon both system characteristics (i.e., number of levels and degree of redundancy) and subject strategy. Of equal importance is the demonstration that the complexity measures related to performance in a time-varying manner, and the nature of this time-varying manner was highly dependent upon events that occurred within the system and the strategy of individual subjects.

D. Conclusions

The experiments, results, and conclusions up to this point have considered the relationship between the design of a large-scale system and human monitoring and control behavior. System characteristics such as number of levels and degree of interconnectedness can have a very strong effect on the ability of humans to maintain proper system operation in the presence of failures. Since the normal system operation tends to be affected in the opposite direction in the presence of the same design characteristics, system designers must be careful to create environments that support both system and human performance.

Straightforward measures were used to assess the complexity of a large-scale system as it relates to the task of monitoring and control. Complexity, as discussed in this report, is a dynamic property of a human-machine system. Complexity varies with time and it varies among operators. Furthermore, complexity is multidimensional: two dimensions of complexity (i.e., structural and strategic) have been proposed and it appears that this distinction is useful both conceptually and practically. Complexity is not due solely to the structure of the system, although a system may certainly be complex due to its structure. Rather, complexity also arises when the human, trying to solve problems within the system's environment, does not understand the structure and as a result issues an inappropriate command, misinterprets display information, etc. In short, systems are also complex due to humans' understanding of the system as reflected by their strategies.

Another result from this work concerns the outcome of complexity. Based on a review of the literature and the major control task of subjects (i.e., finding failures), mean time to failure diagnosis was used as the major dependent measure of complexity. As results suggest, however, mean time to failure diagnosis alone does not completely describe the implications of complexity. For example, the most complex systems resulted in shorter failure-diagnosis times due to the number and location of failures. A smaller fraction of the total number of failures was diagnosed, however. Thus fraction of failures diagnosed was used to explain a different aspect of performance related to task complexity. In short, the result of complexity is multidimensional. A single dimension does not capture the outcome of a complex system.

These comments are important in light of the relationships among system characteristics that contribute to complexity, proper operation of the system, and complexity of monitoring and control by the human. As the system becomes more "complex" (from a nonbehavioral perspective, i.e., more levels and more redundancy), it becomes more resistant to the effects of system failures. Failures take longer to propagate through the more complex systems. Moreover, the effects of any one failure on overall system performance are minimized due to the number of alternate paths through the system. Hence, normal system operation is enhanced. This situation is analogous to the use of redundant or standby equipment in systems to increase fault tolerance. On the other hand, as the system becomes more complex, the task of finding system failures becomes more difficult. Although the system design characteristics can help to avoid the short-term effects of failures, they can have the dual effect of making the human supervisory controller's task more difficult. These findings lend support to Nawrocki's [1981] conjecture that efforts to simplify the task of equipment operation through hardware design tend to complicate the task of equipment maintenance.

The relationship between complexity and human performance takes on increasing importance given the growing prevalence of large-scale systems. Human abilities and limitations in monitoring and controlling these complex systems must be identified in order to design systems that facilitate good failure diagnosis and network management performance. In short, systems must be designed such that they do not overload human information processing capabilities. Beyond the issue of design, an understanding of human performance constraints should facilitate the creation of effective performance aids. Such aids can be used to help people overcome their limitations in coping with the complex environments these systems create, thereby leading to safe and effective system performance.

III. MODELING HUMAN PERFORMANCE

The work described in the preceding sections has implicitly modeled human performance as a function of various system characteristics. Powerful statistical evidence illustrated the strong effect that a system designed for good automatic control can have on a human operator's ability to exercise accurate and timely system intervention. From a behavioral viewpoint, however, the statistical models that have been described do not offer sufficient cognitive explanation for human performance. The empirical analyses describe what happens when humans interact with a large-scale system, but they do not help to explain why things happen that way. Therefore, the second phase of this research program concentrated on the development of a behaviorally valid model of human performance in monitoring and controlling a large-scale system [Henneman 1985b; Zinser 1986; Zinser and Henneman 1986].

Modeling is a good approach in this problem area for several reasons. First, a modeling approach will contribute to a better understanding of human performance in this task. From the previous experiments, much knowledge (both formal and anecdotal) was obtained about how people perform this task. The modeling process allows the formal codification of knowledge and cognitive mechanisms relevant to a complex monitoring and control task. Both human abilities and limitations must be identified by this process. Thus, the model should contain appropriate knowledge representations and implementation mechanisms to provide a high level of behavioral fidelity to human performance.

Second, the modeling approach should facilitate the developmeut of an approach to aiding the human operator. A model that incorporates mechanisms coherent with human cognitive functions should be able to provide meaningful and timely aid to the human operator. Thus, a focus of this work is the use of the model as the basis of an on-line human performance aid.

A. MURRAY : A Model Of Human Problem Solving

The model developed in this report is an extension of a conceptual model of human problem solving proposed by Rouse [1983]. Rouse has suggested that problem solving takes place on three levels: 1) recognition and classification, 2) planning, and 3) execution and monitoring. Thus, when a problem situation develops, the first task is to detect that the problem exists and to categorize it (recognition and classification). An approach or plan to solving the problem must then be developed (planning), and finally, the plan must be implemented (execution and monitoring). The model is further characterized by its ability to make either a state- or a structure-oriented response, depending on both the system state and the human's level of expertise. The model assumes that humans have a preference for pattern-recognition solutions to problems -- that is, humans prefer to make context-specific state-oriented responses to situations. Moreover, the model operates heterarchically at all three problem solving levels almost simultaneously, with situations constantly being re-evaluated relative to their state- or structure-oriented status.

Several efforts have used this generic problem solving model. Domains have included automotive and aircraft powerplants [Hunt and Rouse 1984], process control networks [Knaeuper and Rouse 1985], and communication networks [Viteri 1984]. Performance of these models was, in general, quite good; however, they were constrained by the lack of real "understanding" of the domain by the model. The models lacked knowledge structures that would allow flexible performance strategies to be pursued. Thus, results from efforts at using Rouse's model as the basis for an on-line performance aid [Knaeuper and Morris

1984] were equivocal. A major reason appeared to be the rigidity of the performance strategy of the model.

1. Overview of the Model

The model proposed for the CAIN environment, MURRAY, is illustrated in Figure 5. MURRAY operates in the three stages of Recognition and Classification, Planning, and Execution and Monitoring. Situations are continually reevaluated as system states change due to the system dynamics or operator actions. An important feature of this task is that at any given time the human operator may have several different tasks that could be performed. The key to good performance is the ability to choose among these possibly conflicting subtasks. These model components, their associated representations, and how they interact will be considered below. The section concludes with an example of how the model operates.

2. Knowledge Representation

MURRAY's fidelity to human performance is dependent on the representation of three different types of knowledge needed to perform the task: system knowledge, contextual knowledge, and task knowledge. System knowledge and contextual knowledge are shown explicitly in Figure 5, while the task knowledge is embedded within the Recognition/Classification and Planning components. The Execution component of the model is realized by implementational procedures and the command that is issued.

The first type of knowledge, _system knowledge_, consists of information from CAIN about the current system state, e.g., the number of customers waiting to be served in a city. Thus, the system knowledge of MURRAY is identical to the information presented on the CAIN display. System knowledge is only

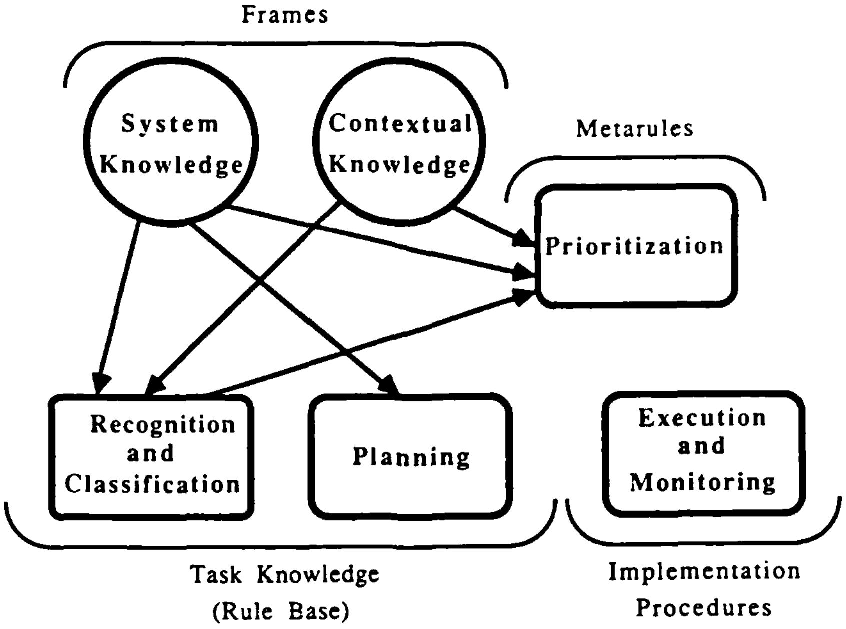


Figure 5. Components of the Conceptual Model

accessed by the model's Recognition/Classification component and the Prioritization mechanism. The system knowledge is structured as a hierarchical frame system [Minsky 1975]. The frame of the highest structural level represents the cluster currently displayed by CAIN. A cluster frame contains information regarding its location relative to other clusters and levels in the network. A cluster frame also contains 16 city frames that correspond to each of the cities (i.e. communication nodes) in the cluster. Each city frame has several "slots" that contain such information as the number of customers waiting for service at the city and the average length of time they have been waiting. These slots are either filled by data that appear on the CAIN display or by appropriate default values. The information contained within this set of frames will change as the information on the display changes. If a new cluster is displayed, the slots in the 16 city frames change to reflect the features of the newly displayed cluster. The slots in the cluster frame will also inherit information from the city of the level above.

The second type of knowledge, contextual knowledge, consists of information concerning the context of the system at a given time, such as locations of individual cities in the network and cities that have high loading and abnormal failure rates. Thus, contextual knowledge is augmented over time; as the model gains ′expertise′, the knowledge stored by this component will change. Contextual knowledge is represented by a network of context frames. This network contains a hierarchy of city and cluster frames as described above, and also data structures that describe both the evolution of the system to the current state, and the human operator′s monitoring behavior and knowledge about the system at any given time. Since the human operator′s knowledge of the contextual features of individual cities (e.g., high failure rate) and the contextual relationships among cities (e.g., Decatur is associated with Atlanta) will vary with time, the model′s contextual knowledge also is augmented as time and, hence, experience increases.

Finally, the third type of knowledge, task knowledge, represents the operator′s behavior in monitoring, problem solving, and failure detection. In other words, task knowledge refers to the knowledge needed by operators to perform their jobs, for example, repairing failed equipment. Task knowledge is represented as a production system [Newell and Simon 1972]. The operator′s heuristics correspond to productions (or rules), while the operator′s internal model of the system corresponds implicitly to ′metarules′ that organize the application of the ′normal′, explicit rules. The metarules are directly implemented in the procedures of the model′s Prioritization component (or inference engine), which will be explained later.

MURRAY contains 22 rules in its representation of task knowledge. These rules have a fixed syntax, and thus, they can be manipulated from outside the program by a text editor. The set of rules is based on a combination of expert judgement and empirical evidence from Experiments One and Two. Each rule consists of a situation and an action part made up of predicates. The situation part of a rule contains one or more predicates. Each predicate may have a value associated with it that relates to either the system or contextual knowledge of MURRAY. The predicates of a rule`s condition part have the function of matching that rule to a CAIN system state or recalled contextual information. Thus, the condition part of a rule corresponds to the Recognition and Classification component of the conceptual model in Figure 5. The action part of a rule contains a command for CAIN. Thus, the identification of a set of potential actions corresponds to the Planning component in the conceptual model.

To summarize, the model depicted in Figure 5 consists of three interacting types of knowledge. System knowledge includes system state information as presented on the CAIN display screen. Contextual knowledge is also acquired from the display, although it is less transient in nature. Contextual knowledge is acquired over time and represents some of the long-term relations among system components. Thus, both system and contextual knowledge can be thought of as forms of declarative knowledge [Anderson 1976]. Task knowledge, on the other hand, is a form of procedural knowledge [Anderson 1976] that delineates how the task should be performed. Details of the implementation of these representations in MURRAY and the way they interact are conaidered in the next section.

3. Implementation

An important part of MURRAY is the inference mechanism of the rule-base representation of the task knowledge. This mechanism determines the way that rules are applied and evaluated. The mechanism is implemented whenever the system state changes, i.e., whenever the model observes a set of new data from CAIN (as a reaction to a command issued by the operator or a dynamic change in the system). At this point, the condition predicates of all the rules are evaluated successively in the Classification component of the model. Those rules whose condition parts match are then prioritized. This prioritization is partially based on *a priori* importance weights that are associated with each rule.

These *a priori* importance weights of applicable rules are dynamically altered by the characteristics of the current system state to which a rule is applied. Fuzzy set theory methodologies are applied to the set of all applicable rules. The use of fuzzy sets can be regarded as a means of representing the phenomenon of activation levels involved in human cognitive processes. (Hunt and Rouse [1984] describe a similar use of fuzzy sets in human performance modeling.) Three factors alter the initial importance weight values to determine the actual importance of a rule in a given situation. These factors are described below.

The first factor is determined by linear fuzzy membership functions. Several of the rules contain fuzzy predicates that describe the current system state in the form of qualitative expressions such as 'high' or 'low'. These values are used to define the membership of the rule in the set of applicable rules. The deviation of the value of the current system state from a 'normal' system state is proportionally weighted by the membership function. Thus, a city with a queue size of 16 customers yields a higher membership function than another city with 12 customers. The membership value always lies between 1 and 2.

The second and third factors are based on memory functions embedded within the model. MURRAY contains two types of memory. The simplest type, on which the second weighting factor is based, is the model's ability to remember previously issued commands. MURRAY is restricted from reissuing the same command within a certain span of time. The factor derived from this kind of memory is a function of time and frequency of usage of a given command. Commands that are used more frequently are more "automatic", and thus, are retained for less time in memory [Henneman and Rouse 1984c]. The numerical value resulting from this factor is also presented to the prioritization of a given rule and always lies between 0 and 1. The more recently a command was issued, the lower is the value of this second factor. (A value of 1 represents the fact that memory retrieval for a given command failed.)

The second type of memory, which is more complicated than the simple command memory described above, is the contextual knowledge that an operator accrues over time due to learning. This type of memory is the basis for the third rule weighting factor. The factor is bounded between 1 and 2 and its value increases with contextual representativeness. This form of memory is implemented by the context frame structures that were explained earlier and accessed by the inference mechanism upon application of rules that allow actions to be activated from the contextual memory instead of solely from a system state. Information in the context frames is updated whenever a particular city is displayed. If no updates occur, the retention of the context frame decreases over time until it is eventually deleted from the contextual memory.

The dynamic importance of a rule in the given context is finally obtained by multiplying the a priori weight value of that rule by each of these three factors. The rule that is eventually selected is determined by ordering the applicable rules in descending order of derived importance in a priority queue and using a head-of-the-line queueing discipline. The first element of the priority queue is the model´s first choice of the next CAIN command. The following section presents an example of how this mechanism works.

4. Example

Consider a situation in which the currently displayed cluster is at the highest level of the system. The previously observed cluster was on the second system level below Chicago and the operator´s last command that was issued was an ´up´ command. Information available in the cluster frame structure includes the queue sizes of the 16 top level cities and their locations. For simplification, only three cities with the largest queue sizes will be considered (Seattle(17), Chicago(14) and Atlanta(15)). Previously, the cluster below Boston had a large number of customers and Dubuque (in the level below Chicago) indicated a high queue size. This information was retrieved from the model´s contextual memory. Also, the cluster below Seattle was recently displayed. Both the ´monitor´ and ´test´ commands were issued recently. The watch list of cities with recurring failures was also observed fairly recently and the only city on it was Houston.

The rules applicable to this situation (as derived by the Recognition/Classification component of the model) are listed in Table 3. The numbers in brackets represent the importance weight, the factors derived from the fuzzy functions (fuzz), from the simple command memory (cmd), and from the contextual memory (ctxt), and finally the overall dynamic importance of the rule is listed. All the values of the three weighting factors can be explained by the above given information about the situation. For example, the dynamic importance for ´down (Chicago)´ is obtained by its current queue size of 12 (1.20), the fact that the same command was given before (0.60) and its contextual situation (1.10).

During the prioritization process, ´down (Seattle)´ is initially selected as a command (*). The next several possibilities are not considered as commands since their final importance is less than 72. The ´test´ command ultimately yields the highest priority in the given context (**), and thus, is implemented as the next CAIN command. It is interesting to see how close some of the prioritization decisions are. This phenomenon (which suggests that in a given situation more than one action may be ´correct´) will be further discussed in the next section when validation issues of the model are addressed.

	Command	Weighting Factors				Final Weight	
			fuzz	cmd	ctxt		
#2	down(Seattle)	[47	1.70	0.90	1.00	72]	(*)
#2	down(Chicago)	[47	1.20	0.60	1.10	37]	
#2	down(Atlanta)	[47	1.50	1.00	1.00	70]	
#3	mon	[36	1.00	0.95	1.00	34]	
#6	test	[84	1.00	0.87	1.00	74]	(**)
#11	mon	[44	1.00	0.95	1.00	42]	
#20	down(Boston)	[25	1.10	1.00	1.30	36]	
#21	down(Dubuque)	[24	1.05	1.00	1.20	30]	
#22	down(Houston)	[21	1.12	1.00	1.15	27]	

Table 3. A Fuzzy Set of Rules

B. Experiment Three: Model Evaluation[4]

1. Method

Experimental data were collected to validate MURRAY. Ten junior and senior Industrial Engineering majors participated in a total of nine sessions (3 training, 6 data collection) monitoring and controlling CAIN. Subjects read 2 sets of written instructions that described CAIN and its operation prior to Sessions 1 and 2. At the end of Session 3, subjects took a quiz to verify their knowledge of CAIN and to assess their level of contextual knowledge. Subjects took a similar quiz at the end of Session 9. Each session lasted approximately 45 minutes and subjects were paid $50.00 for their participation.

Independent variables considered in the experiment were Session (6 levels) and Subject (10 levels). Session was of interest to assess if subject or model performance (and level of agreement between the two) improved or degraded with time. Individual subject performance was of interest to assess if degree of model-subject agreement was a function of individual strategy differences.

Comparison of MURRAY and subject performance was done in two ways. First, an "open-loop" comparison was made in which subject performance was compared with MURRAY's performance. Second, a "closed-loop" analysis was performed. Subject data files were replayed concurrently with a version of MURRAY. Whenever a subject action was performed, MURRAY generated the action it

[4] Results described in this section are reported in greater detail in Zinser and Henneman [1987a] and Zinser [1986].

would implement, along with a list of its other applicable rules. The subject´s action was then implemented. This form of analysis allowed an action-by-action (or process) performance comparison to be made.

2. Open-loop Evaluation

Three measures, mean customer sojurn time, number of customers served, and fraction of failures repaired, were used to assess subject performance. Table 4 contains values for each of these measures for each subject, averaged across all six experimental sessions and ranked in order of improving performance quality. MURRAY outperformed all of the subjects in terms of mean sojurn time and number of customers served. MURRAY ranked eight, however, in terms of the third performance measure, fraction of failures repaired. There was not, however, any statistical difference between MURRAY´s and subjects´ per-

Mean Sojurn Time (sec)		Customers Served		Fraction of Failures Repaired	
Mean	Subj	Mean	Subj	Mean	Subj
80.66	5	4623	5	49.55	3
80.50	1	4626	4	51.85	8
80.50	3	4632	3	54.35	4
80.33	4	4686	8	55.52	MURRAY
80.00	8	4712	1	55.55	10
77.50	10	4756	10	58.37	7
77.16	9	4782	9	59.41	5
77.16	2	4831	2	59.96	9
76.50	7	4835	7	59.97	6
76.16	6	4840	6	60.05	1
75.33	MURRAY	4880	MURRAY	61.44	2

Table 4. Subject and MURRAY Performance

formance as measured by Duncan's multiple range test.

As Henneman and Rouse [1986] have noted, the measure of fraction of failures repaired may conceal other important aspects of performance. For example, as described previously, CAIN system failures occur in three different ways: randomly occurring failures, capacity failures, and recurring (or non-random) failures. It may be argued that an indication of the operator's ability to maintain a stable system is given by the number of capacity failures that occur; the frequency of other failure types is really out of the operator's control. The number of capacity failures incurred by subjects and averaged across sessions resulted in 22 failures; MURRAY, however, allowed only 13 capacity failures to occur. Thus, despite the lower fraction of failures found by MURRAY, the model-operated system actually allowed fewer failures in total to occur, thus resulting in more stable system performance.

Subject and MURRAY performance was also compared based on individual command usage. Table 5 shows the percent of times individual commands were used by subjects and MURRAY. This comparison shows a high degree of similarity; major differences involved MURRAY's preference for monitoring the system. This activity resulted in much information being displayed. Subjects were apparently reluctant to ask for all of this information, whereas MURRAY could easily process all of these data. A comparison of command sequences showed similar results: MURRAY tended to favor commands that would generate the most information on which to base future actions. Table 6 compares the percent of times certain command sequences were used by subjects and MURRAY.

These results from the open-loop analysis suggest that MURRAY achieved a high quality of performance that appears to be consistent with that of human subjects. Nevertheless, it is clear that differences must exist that can

SUBJECT	m	w	d	u	a	t	r	l
1	7	1	27	3	13	40	10	0
2	8	2	26	5	10	43	6	1
3	11	3	22	0	12	37	12	3
4	11	3	18	15	2	39	12	0
5	9	1	28	2	13	42	6	0
6	10	1	31	3	10	40	6	0
7	16	1	25	4	11	37	7	0
8	6	1	23	13	4	38	16	0
9	2	1	36	1	7	43	10	0
10	8	1	27	3	14	40	8	0
AVERAGE	8.8	1.5	26.3	4.9	9.6	40	9.3	0.4
MURRAY	13	0	27	14	7	35	4	0

Table 5. Command Use by Subjects and MURRAY (Percent of Total)

		→1st command									→1st command						
		m	w	d	u	a	t	r			m	w	d	u	a	t	r
↓	m	0	0	0	0	1	6	0	↓	m	0	0	0	0	0	13	0
	w	0	0	0	0	0	1	0		w	0	0	0	0	0	0	0
	d	7	1	3	1	1	13	4		d	11	0	4	2	1	8	2
	u	0	0	0	0	0	4	1		u	3	0	2	0	0	9	1
2nd	a	0	0	1	0	0	8	1	2nd	a	0	0	3	0	0	4	0
cmd	t	1	0	26	3	8	1	1	cmd	t	0	0	18	12	5	0	0
	r	0	0	0	0	0	7	2		r	0	0	0	0	0	3	0

Table 6. Command Sequences Issued by Subjects and MURRAY (Percent of Total)

account for the discrepancies between MURRAY and human performance. Reasons for these differences are explored in the following section.

3. Closed-Loop Evaluation

Subject and MURRAY performance was also compared on an action-by-action basis, thereby facilitating a comparison of subject and model in exactly the same environmental conditions. This type of analysis allows a validation of the behavioral processes and representations present in the model. Matches were differentiated in three different ways: Type I - same command issued by MURRAY and subject at same time; Type II - same command, different time (i.e., one command earlier or later); Type III - commands that belong to the same class, issued at the same time (e.g., a ′down′ command to different clusters). In addition, subject commands were compared to the first three choices that MURRAY had listed in its priority queue of applicable actions. By averaging across subjects and sessions, MURRAY matched subjects′ actions across Type I, II, and III matches 75.3% of the time. MURRAY exactly matched (Type I matching) subject performance 64% of the time. These results are impressive given the subjective nature of the rule identification method and development of prioritization weights.

The results for each experimental session were analyzed with ANOVA. The difference between sessions was not significant, whereas the difference between subjects was highly significant ($p<0.0001$). This result suggests that some subjects used different strategies from the model but did not change them over time.

A critical decision that subjects must make in this task is when a new cluster should be displayed and which one it should be. One reason for the 25% of the commands not explained by the model was found by comparing the frequency of matches with just the ′down′ (i.e., ′change screen′) commands. Considering only the first three of MURRAY′s choices at every iteration, subject performance (with respect to only ′d′ commands) agreed just 12% of the time (Type I matching). The differences between subjects were significant ($p<0.0001$), but there were no significant differences between sessions. This result is consistent with the findings of Henneman and Rouse [1986]: humans not only use symptomatic and topographic search strategies, but also use serendipitous and other random-appearing search strategies. These strategies are neither represented nor supported by MURRAY in its task knowledge. Nevertheless, by relying on its task description provided in the rule base, MURRAY resulted in uniformly excellent performance. Therefore, a model-based aid might be useful in providing the operator with procedural instructions; MURRAY could support the operator with additional or alternative strategies to monitor or control CAIN. In addition, MURRAY could provide support in accessing the network by identifying problem areas that are most critical.

C. Conclusions

To summarize, MURRAY proved to be a reasonable means of describing human behavior in a complex monitoring and control task. Open-loop analysis of model performance indicated that the model consistently did as well as human operators. Closed-loop, action-by-action comparison of subject and MURRAY performance revealed a high degree of behavioral congruence. Thus, it appears that the structures and mechanisms present in the model produce quite similar behaviors to humans′ structures and mechanisms used in performing this task.

Nevertheless, it should be noted that the level of matching was not perfect. Both MURRAY and human operators appear to have different strengths that are useful in this environment: MURRAY is good at prioritizing tasks; the human operator is good at improvising flexible search strategies. Thus, a combination of the two could result in improved overall system performance. The next step in this research program, therefore, was to implement a human performance aid based on MURRAY. Such an aid should provide cognitively plausible assistance to the human operator.

IV. AIDING HUMAN PERFORMANCE

Aiding human performance in a system may be done in many ways. For example, it may be possible to aid human performance simply by altering the characteristics of the display of information to the human operator [Mitchell and Saisi 1986]. Alternatively, the aid may provide advice to the human based on some normative representation of a task, such as multi-attribute utility theory [Freedy, et al. 1985]. Still other approaches may use system simulation to allow the human operator to ask "what if" questions of potential actions [Yoon and Hammer 1986]. Coupled with decisions regarding the selection of an appropriate aiding scheme are decisions concerning task allocation. For example, if an aid is able to suggest appropriate operator actions, it might be acceptable to allow the aid to implement its own suggestions in some situations.

In the context of CAIN, one can imagine potential operator performance aids. A simple alteration of the displays (e.g., highlighting the most salient visual cues) could likely lead to a performance improvement. Another approach might be based on the complexity measures described earlier: the aid could make recommendations based on actions that would reduce complexity by the greatest amount. In this section, one particular approach to aiding the human operator is developed and evaluated. The approach proposed here is based on the model of human operator performance, MURRAY, that was discussed in the preceding section. Since the model contains knowledge structures and mechanisms congruent with those underlying humans' behaviors, the model should be effective in providing meaningful advice to the human operator [Knaeuper and Morris 1984]. Thus, the model-based aid evaluated here is significantly different from the decision support available from expert systems or other normative approaches. Although MURRAY's advice is always derived from a set of if-then rules (as is an expert system), MURRAY's decisions are based on its embedded knowledge structures (i.e., contextual and system) and its prioritization mechanism to resolve conflicts among rules. The model is only expert in the sense that it makes use of all available information on the complex CAIN display, has a good memory, is not pressured by time-critical situations, etc.

The implementation of the MURRAY-based aid is largely one of designing an appropriate interface. The design of this interface is critical in that the operator should be neither overloaded with information nor preoccupied with requesting advice. In view of the complexity of the existing CAIN display and associated operator functions, the decision was made to implement a simple, straightforward interface for the aid. The mechanism works as follows. MURRAY operates in real-time in parallel with the human operator who is controlling the system. MURRAY suggests a single command to the operator upon request, i.e., whenever the operator issues an 'h'-command ('help'). MURRAY's highest ranked choice for the next command is presented on the CAIN display next to the command entry line at the lower center part of the display (Figure 2). Considering factors such as the operator's mental workload and the time-constrained dynamic environment, this rather simple augmentation of the display was selected over other possible implementations, such as multiple command options, displaying further information relative to MURRAY's prioritization process, or even adding another display with aiding information. This interface is directed at the expert end user (such as the CAIN operator) rather than a sporadic novice user.

A. Experiment Four: Aid Evaluation

1. Method

The evaluation of the on-line aid was performed by augmenting Experiment Three described in the previous section. The main goal of Experiment Four was to assess the effects of on-line aiding on operator performance in the CAIN environment. The experimental design used to evaluate this issue was a between-subjects design in each of two treatment groups: unaided (using the subject performance data from Experiment Three) and aided operation of CAIN (a new group of 10 subjects). Thus, the treatment structure is a one-way factorial design with aiding being the independent variable of interest.

A second group of ten paid subjects participated in operating CAIN for nine sessions. Instructions, training, and questionnaires were presented in three sessions as in Experiment Three. The difference in this experimental condition was the availability of the on-line aid. The instructional material was augmented by a description of the ´help´ command. The new command was introduced in the second training session. The subjects were instructed to use the aid when uncertain about what to do next or to enhance their own strategies. Subjects were also told to implement the aid´s suggestion only if they felt it was reasonable.

2. Results[5]

From several perspectives, the aid had no impact on subject performance. Table 7 compares aided vs. unaided subject performance for the three

[5] Results reported in this section are reported in more detail in Zinser and Henneman [1987b] and Zinser [1986].

performance measures of mean customer sojurn time, number of customers served, and fraction of failures repaired. The table lists performance averaged across all subjects for each group (aided or unaided) as well as the performance of the best and worst subject in each group. ANOVA revealed no statistical differences between groups on the various measures of subject performance. Although the aided group performed slightly better on the average in terms of mean customer sojurn time and number of customers served, the unaided group performed slightly better in terms of fraction of failures found. Interesting to note, however, is that the performance of the best and worst subjects is always better for the aided group. A comparison of command usage also showed no major systematic differences between groups.

At first observation, these results are disappointing. However, a more fine-grained analysis of the data revealed ways in which the aid was quite helpful. For example, although there were no statistical differences between groups, aided subjects were able to find certain types of failure faster than unaided subjects, thus maintaining a more stable system. Table 8 shows the mean time to repair for aided and unaided subjects for all types of failure.

	AVERAGE		BEST		WORST	
	Aid	No aid	Aid	No aid	Aid	No aid
Sojurn Time (sec)	78	79	75	75	84	88
No. of customer served	4733	4729	4892	4884	4451	4447
Fraction failures found	.56	.57	.65	.62	.49	.48

Table 7. Aided vs. Unaided Subject Performance: Mean Performance Measures

Failure type	Aid	No aid
Total	39.4	39.6
Random	96.5	106.7
Capacity	23.5	29.6
Non-random	24.0	23.6

Table 8. Aided vs. Unaided Subject Performance: Mean Time to Repair (sec)

In particular, the aided subjects found both random and capacity failures more quickly than the unaided subjects. Accordingly, unaided subjects had more failures occur during their experimental sessions. Table 9 shows the average number of failures that occurred of each type for each experimental group. In each category, the aided group had fewer failures. This trend was strongest for the capacity failures, which, as noted previously, are an indication of system stability.

In addition, the aid enjoyed a high level of acceptance by subjects. On the average, 83% of all commands that were suggested by the aid were actually implemented by subjects. Given that aid requests constituted only 8% of all commands issued, however, this high level of acceptance was not reflected in

	Aid	No aid
Total	72	75
Random	30	30
Capacity	21	22
Non-random	20	23

Table 9. Aided vs. Unaided Subjects Performance: Mean Number of Failures

the overall performance scores. These results suggest that more emphasis should be given in the future to training operators in the use of the aid to illustrate its benefits. Finally, the questionnaires completed by subjects at the end of the experimental sessions indicated that aided subjects had a higher level of contextual knowledge (as measured by number of second-level city locations correctly recalled) than unaided subjects. Aided subjects correctly identified the location (on the average) of 26 cities; unaided subjects recalled the location of 22 cities.

As emphasized in Section III, one of the strengths of MURRAY is its ability to prioritize tasks. In fact, a key to good performance in this task is the ability to decide which part of the network should be observed next. It is interesting to note, therefore, that the percentage of times a ′d′-command suggested by the aid led to finding a failure was 34%; the percentage of times any ′d′-command issued by a subject that was not suggested by the aid led to finding a failure was only 10%. Clearly, from this perspective the aid was quite beneficial in providing useful aid to the operator. Nevertheless, since the aid was requested infrequently, these fine-grained results were not reflected in the overall performance scores. As mentioned above, the low level of use masked any overall performance improvement.

B. Conclusions

In the final phase of this research program, an on-line performance aid based on MURRAY for human monitoring and control in a large-scale system was introduced, described, and evaluated. Model-based on-line aiding was selected because previous efforts have shown its potential benefits [Knaeuper and Morris 1985]. Experimental results, however, failed to show significantly improved overall performance of aided subjects. Nevertheless, more fine-grained evaluation of the results demonstrated subtle subject improvement in some performance aspects. One of these aspects was a more stable operation of the CAIN system by aided subjects. The second and most important result was improved subject performance in the critical decision of selecting which part of the network to observe next. The aid provided clear performance improvement with respect to failure-detection strategies. These subtle performance improvements suggest that further research is needed to determine if alternative implementations of the aiding approach could result in more definitive results.

Several other aiding approaches are viable given the experimental results presented in this report. For example, the fact that subjects did not request the aid very often suggests that different results could be obtained if the model's suggestion was always available. (Such an approach would be consistent with the model-based aid used by Knaeuper and Morris [1984]). Alternately, the aid could present its recommendation only if the derived importance ranking of the rule exceeded some threshold value. A related approach would be to emphasize the use of the aid through training as mentioned above. Another alternative would be to alter the strategy of the model so that it would support problem-solving strategies significantly different from the human's. Zinser [1986], for example, found that if the *a priori* weightings of the model's rules that were related to contextual knowledge were increased, command matches with human performance decreased; the model began to place more emphasis on context-dependent strategies (e.g., recalling that Dubuque has recurring failures). Given that approximately half the failures in CAIN were dependent on the context, a strategy based more on contextual recall and augmented with 'normal' human strategies should be very effective. In light of the ambiguity of the current results, these ideas merit careful further consideration.

V. CONCLUSIONS

The research described in this report has considered human performance in the monitoring and control of large-scale systems from many perspectives. Initial efforts empirically examined the effects of system design parameters on human performance. The results clearly illustrated the problems that people have in controlling a multiple-level system. Large performance differences were noted when the number of system levels increased from two to three. Multiple-level systems tend to mask failure symptoms from the human operator. Although such systems protect upper system levels from the effects of failures, when the failures do propagate upwards, their effects are more serious. Unfortunately, the results presented here indicate that people tend to wait until symptoms emerge rather than pursuing a more active search strategy.

Similar comments can be made regarding the degree of redundancy present in a system. Increased *connectivity* between system parts led to improved automatic system performance but degraded human failure-diagnosis performance. Thus, system designers need to be aware of the tradeoffs that can be made between supporting automated system control and human failure-diagnosis performance. Moreover, if the physical structure of the system cannot be altered to support good human performance, then aids must be designed within the system to cause the human operator to adopt effective control strategies.

Other human limitations in dealing with large multiple-level systems were also noted. For example, when contextual information was introduced to the system, humans did not adopt strategies that took any great advantage of this information, even though they were aware of certain types of failure that could be located more readily by using contextual knowledge. Humans used a rather mechanical strategy that did not rely on the context. People also had difficulty in prioritizing subtasks in time-critical situations. It was shown that by relying on a model-based aid with a very good prioritization method the human could make better search decisions. Again, people tended to learn one way of performing the task and not change as the environmental conditions shifted.

The notion of relying on an aid based on human cognitive functions deserves much closer scrutiny. Despite some ambiguity, the results discussed in this report are promising: subtle performance improvements were shown when the aid was used by subjects. The approach is consistent with the views espoused by Rasmussen [1985] regarding the support of human operators in complex systems. In particular, Rasmussen argues that an aspect to consider in the design of a system is "a representation of the information processing capabilities and limitations of the decision maker and of the subjective formulation of goals and criteria for choice among possible strategies..." MURRAY provided such a representation of the human operator that was shown to be of use in decision support. MURRAY gave "cognitively plausible" advice to the human operator when that information was needed.

Nevertheless, the CAIN system (as augmented with MURRAY's advice) is limited in the support it can provide the operator. Rasmussen [1985] argues that systems should support an operator at various levels in an abstraction hierarchy of functions and according to various levels of aggregation. Although CAIN does support various levels of aggregation, CAIN's level of abstraction to the human operator is fixed. Future work should concentrate on defining the system functions and representations at various levels of abstraction.

Concurrent efforts should be directed at considering some of the issues related to aiding mentioned in Section IV. In particular, alternate interface design methods, task allocation strategies, and issues related to user acceptance should be considered. Also, the notion of implementing a model with search strategies complementary to (as opposed to coincident with) human strategies (e.g., based on the contextual information) should be explored. Aids based on models complementary to human strategies may have more potential to improve overall performance but may be difficult for the human operator to understand. On the other hand, aids based on models coincident with human strategies may be easy for the human to understand but may not enable any improvement over unaided performance. This potential trade-off deserves more consideration.

In summary, the material presented here has made several important contributions. First, it has added to a general understanding of the relationship between system design characteristics and human performance. Second, from a theoretical perspective, this project has contributed a framework for measuring the complexity of a system based on the physical system characteristics and the human's understanding of these characteristics. Third, a model of human performance was proposed and evaluated that was made up of several different interacting knowledge structures and cognitive mechanisms. The model was shown to produce behavior consistent with human performance. Finally, this model was shown to be effective as a means of aiding human performance in a complex monitoring and control task.

REFERENCES

American Telephone and Telegraph. Connecting a nation. Administrative Training Center, Morristown, NJ.

American Telephone and Telegraph. Notes on the network: Section II. Administrative Training Center, Morristown, NJ.

Anderson, J.R. Language, Memory, and Thought. Hillsdale, NJ: Lawrence Erlbaum, 1976.

Ash, G.R. and V.S. Mummert. AT&T carves new routes in its nationwide network. AT&T Bell Laboratories Record, August 1984, pp. 18-22.

Box, G.E.P. and G.M. Jenkins. Time series analysis: forecasting and control. Holden-Day: San Francisco, 1976.

Freedy, A., A. Madni, and M. Samet. Adaptive user models: methodology and applications in man-computer systems. In W.B. Rouse (Ed.), Advances in Man-Machine Systems Research (Vol. II). Greenwich,CT: JAI Press, Inc. 1985.

Henneman, R.L. Human problem solving in large scale dynamic networks. Ph.D. Dissertation, Georgia Institute of Technology, CMMSR Report No. 85-1, 1985a.

Henneman, R.L. A model of human performance in a large scale system. Proceedings of the International Conference on Cybernetics and Society, Tucson, AZ, 1985b, pp. 527-531.

Henneman, R.L., and W.B. Rouse. Human problem solving in large scale networks. Proceedings of the IEEE International Large Scale Systems Symposium, Virginia Beach, VA, 1982, pp. 293-297.

Henneman, R.L., and W.B. Rouse. Human performance in monitoring and controlling hierarchical large scale systems. Proceedings of the Human Factors Society 27th Annual Meeting, Norfolk, VA, 1983, pp. 685-689.

Henneman, R.L., and W.B. Rouse. Measures of human performance in fault diagnosis tasks. IEEE Transactions on Systems, Man and Cybernetics, SMC-14, (1):99-112, 1984a.

Henneman, R.L., and W.B. Rouse. Human performance in monitoring and controlling hierarchical large scale systems. IEEE Transactions on Systems, Man and Cybernetics, SMC-14, (2):184-191, 1984b.

Henneman, R.L., and W.B. Rouse. Assessing the complexity of a large scale system: measures of system structure and human strategy. Proceedings of the 1984 IEEE International Conference on Systems, Man and Cybernetics, Halifax, NS, 1984c, pp. 68-72.

Henneman, R.L., and W.B. Rouse. On measuring the complexity of monitoring and controlling large scale systems. IEEE Transactions on Systems, Man and Cybernetics, SMC-16, (2):193-207, 1986.

Hunt, R.M., and W. B. Rouse. A fuzzy rule based model of human problem solving. IEEE Transactions on Systems, Man and Cybernetics, SMC-16, (2):112-119, 1984.

Keeney, R.L. and H. Raiffa. Decision making with multiple objectives. New York: Wiley, 1976.

Knaeuper, A. and W.B. Rouse. A rule based model of human problem solving behavior in dynamic environments. IEEE Transacticns on Systems, Man and Cybernetics, SMC-15, (6):708-718, 1985.

Knaeuper, A. and N.M. Morris. A model based approach for on-line aiding and training in process control. Proceedings of the 1984 IEEE International Conference on Systems, Man and Cybernetics, Halifax, NS, 1984, pp. 173-177.

Minsky, M. A framework for representing knowledge. In P. Winston (Ed.) The Psychology of Computer Vision. New York: McGraw Hill, 1975.

Mitchell, C.M. and D.L. Saisi. Use of model-based qualitative icons and adaptive windows in workstations for supervisory control systems. Submitted for publication, 1986.

Mocenigo, J.M. and D.M. Tow. Managing a network that won´t sit still. AT&T Bell Laboratories Record, August 1984, pp. 23-26.

Nawrocki, L.H. Computer-based maintenance training in the military. In J. Rasmussen and W.B. Rouse (Eds.) Human Detection and Diagnosis of System Failures. New York: Plenum Press, 1981.

Newell, A. and Simon, H.A. Human Problem Solving. Englewood Cliffs, NJ: Prentice Hall, 1972.

Paap, K.R. and R.J. Roske-Hofstrand. The optimal number of menu options per panel. Human Factors, 28, (4):377-385, 1986.

Rasmussen, J. The role of hierarchical knowledge representation in decision making and system management. IEEE Transactions on Systems, Man and Cybernetics, SMC-15, (2): 234-243, 1985.

Rouse, W.B. Models of human problem solving: detection, diagnosis and compensation for system failures. Automatica, 19, (6): 613-625, 1983.

U.S. Army. Combat communications within the division. Field Manual 11-50, March 1977.

U.S. Army. Communications-electronics management system. Field Manual 24-22, June 1977.

Viteri, E. A rule based model of a human operator in a complex communication network. M.S.I.E. Thesis, Georgia Institute of Technology, CMMSR report no. 84-3, 1984.

Wickens, C.D. Engineering Psychology and Human Performance. Columbus, OH: C.E. Merrill Publishing Company, 1984.

Yoon, W. and J.M. Hammer. Aiding the operator during novel fault diagnosis. Submitted for publication, 1986.

Zinser, K. Modeling and aiding human decision making in operating and supervising dynamic systems. M.S.I.E. Thesis, Georgia Institute of Technology, CMMSR report no. 86-4.

Zinser, K. and R.L. Henneman. Evaluation of a model of human performance in a large scale system. Proceedings of the 1986 IEEE International Conference on Systems, Man and Cybernetics, 1986, pp. 869-874.

Zinser, K. and R.L. Henneman. Development and evaluation of a model of human performance in a large-scale system. Submitted for publication, 1987a.

Zinser, K. and R.L. Henneman. A model-based aid for human performance in monitoring and controlling a large-scale system. Submitted for publication, 1987b.